

# LES PROCESSUS CRÉATIFS DANS LA LUTHERIE NUMÉRIQUE : ANALYSE D'ELLE DE LORENZO BIANCHI HOESCH

João Fernandes

Université Paris 8, MUSIDANSE

joaoedufernandes@hotmail.com

## RÉSUMÉ

Cet article vise à montrer le rôle de la lutherie numérique dans la création d'une identité sonore d'un musicien-luthier. Pour comprendre ce travail, l'instrument de musique numérique sera disséqué et chacun de ses éléments sera expliqué. Ce texte souhaite montrer comment les différentes façons d'aborder la construction de chacun des composants des instruments numériques ont des conséquences pratiques dans la musique produite. Ces prémisses seront confirmées par l'analyse d'un cas pratique : *Elle*, l'instrument de musique numérique du compositeur Lorenzo Bianchi Hoesch. Dans cet article, il s'agira de regarder les aspects principaux dans la conception de l'instrument *Elle* permettant de comprendre la pensée du musicien. On verra comment les décisions prises dans l'élaboration de l'instrument impriment dans sa musique des caractéristiques dont le musicien ne peut pas se dissocier. Enfin, on montrera comment le changement d'un élément constitutif de l'instrument implique plusieurs changements dans son expressivité.

## 1. INTRODUCTION

Une caractéristique des nouvelles lutheries numériques est la fusion entre le rôle de luthier et de musicien. Très souvent, ce sont les musiciens eux-mêmes qui développent leurs instruments en fonction de leurs besoins expressifs [8]. Au lieu d'apprendre simplement à jouer un instrument de musique préexistant, ces musiciens-luthiers peuvent construire leurs instruments en valorisant leurs compétences acquises précédemment. De même, l'instrument peut répondre plus adéquatement aux exigences esthétiques du musicien. À noter que les communautés de développement autour des logiciels partagent beaucoup d'informations : les modules audio et leur code de programmation. Cela permet un accès facile à ces informations et un apprentissage rapide dans le travail du luthier [5].

Dans cet article, je me propose de montrer un cas d'étude où le musicien a créé son propre instrument de musique numérique. La première partie du texte montre les différents éléments qui composent un instrument numérique et ensuite j'analyserai l'instrument *Elle* du compositeur, improvisateur et artiste sonore Lorenzo Bianchi Hoesch<sup>1</sup>. Cette deuxième partie décrit certains des choix importants du musicien-luthier lors de la conception de son instrument. Pour ce faire, une description

de ses différents éléments est nécessaire pour ensuite tirer des conclusions sur le résultat sonore qui émerge de ces choix. J'ai choisi d'analyser l'instrument de cet artiste, car son utilisation est très éclectique. L'instrument *Elle* est utilisé à la fois pour l'improvisation électroacoustique et pour la composition de pièces. Il est utilisé pour faire des performances en solo ou comme un instrument d'ensemble. L'instrument *Elle* n'est pas seulement employé par Bianchi Hoesch, mais aussi par plusieurs de ses étudiants, dont certains collègues improvisateurs de l'auteur de cet article.

## 2. L'INSTRUMENT DE MUSIQUE NUMÉRIQUE

Les nouveaux outils et appareils électriques, électroniques et digitaux utilisés pour la création musicale amènent à réfléchir sur les catégories existantes pour saisir le rapport à ces objets. Une définition générale d'instrument de musique englobant les instruments numériques nous est donnée par Bruno Bossis : « Un instrument de musique ne peut être considéré comme tel que s'il comprend la chaîne complète du geste à l'émission sonore ou au moins à un signal électrique analogue à cette émission. [...] Un ordinateur n'en est pas un, mais s'il est muni d'une entrée son ou d'une interface gestuelle, d'un logiciel de synthèse ou de traitement, alors l'ensemble du dispositif peut être considéré comme un instrument de musique. » [1].

La définition de Bossis montre déjà que le musicien avec ses actions (le geste) manipule une « interface » pour commander un « moteur sonore » (logiciel de synthèse ou traitement numérique) pour être projeté dans un espace. Il faudra ajouter à cette définition l'unité qui permet la traduction et l'envoi des données d'un élément à l'autre. Les instruments musicaux numériques sont donc composés d'un moteur sonore, d'une interface gestuelle et de cette unité abstraite pour faire la connexion entre les données des deux unités, souvent appelée lien instrumental (ou *mapping*). Le moteur sonore est créé avec un logiciel dans un ordinateur (avec ses composants : carte son, clavier, souris, moniteur, etc.), muni d'une unité de DSP (*digital sound processing*). Chaque élément devrait aussi comporter un module pour envoyer une rétroalimentation à l'instrumentiste. Pour le moteur sonore, il s'agit du système de diffusion.

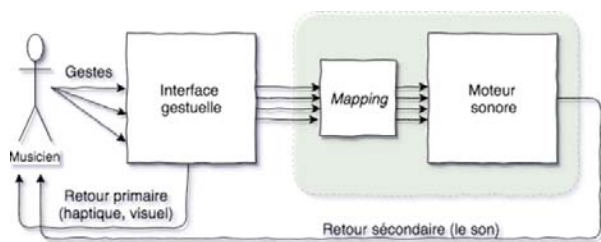
Pourtant, malgré cette définition générale, les luthiers tendent à définir l'instrument numérique à partir de la composante qu'ils privilégient dans leur travail [6]. Par exemple, Thor Magnusson définit volontairement

<sup>1</sup> Voir [brahms.ircam.fr/lorenzo-bianchi-hoesch](http://brahms.ircam.fr/lorenzo-bianchi-hoesch).

l'instrument numérique comme « *a theoretical construction where all knowledge of its properties, as a sound or pattern generating tool, has to be consciously and symbolically inscribed into the instrument in the form of written code* » [7]. La question de la composition de l'instrument le mène à donner plus d'importance au moteur sonore et au code : « *This leads to the definition of the instrument as primarily a combination of a sound engine and a mapping layer where the gestural interface is of secondary importance* » [7]. On verra plus tard que cette définition en particulier s'applique au cas de l'instrument *Elle*.

À l'inverse, l'équipe d'ACROE (Association pour la création et la recherche sur les outils d'expression) pense l'instrument numérique de la même manière que les instruments acoustiques. Dans leur cas, la technologie sert à reproduire l'instrumentalité existant dans les instruments acoustiques de façon à ce que le couplage instrumental entre le musicien et l'interface gestuelle se réalise de manière analogue.

Un schéma d'un instrument numérique peut être observé dans la Figure 1. Du point de vue de l'instrumentiste, les trois éléments mentionnés avant se résument à un seul, mais il est important de souligner que chaque unité isolée ne compose pas l'instrument. C'est dans l'interaction entre ces trois éléments que l'instrument de musique numérique émerge.



**Figure 1.** Structure habituelle d'un instrument de musique numérique

Je regroupe le moteur sonore et le *mapping* dans le carré gris, car ils sont habituellement réalisés dans l'unité de calcul du système. Toutefois, actuellement, le *mapping* peut être réalisé directement dans l'interface gestuelle, car souvent celle-ci possède aussi une unité de calcul.

Dans cet article il ne s'agit pas de faire une description détaillée de chaque élément, puisqu'il existe une vaste bibliographie autour du sujet. On donnera rapidement des définitions pour que le lecteur puisse se situer dans l'analyse d'*Elle*, l'instrument numérique de Lorenzo Bianchi Hoesch.

### 2.1. L'interface gestuelle

Il s'agit de l'élément de l'instrument avec lequel le musicien interagit. Il est responsable de la traduction des actions de l'instrumentiste dans des valeurs compréhensibles par la machine et de leur envoi à l'interface de *mapping*. Plusieurs dénominations sont proposées pour cet élément de l'instrument : contrôleur gestuel (*gestur-*

*al controller*), dispositif de saisie (*input device*), surface de contrôle.

Une définition nous est donnée par Mulder : « *A control surface consists of sensors for human movement capture, actuators for tactile as well as force feedback yielding a haptic representation and last but not least a visual representation. The control surface outputs data that represents the movements and gestures, which data is turned into sound variations after processing.* » [9]

### 2.2. Le moteur sonore

Cet élément de l'instrument numérique est l'unité responsable de la production sonore. L'interface de contrôle transforme le geste en information numérique, qui à travers le lien instrumental, l'envoie au moteur sonore qui s'occupe de produire le son. Souvent réalisé dans un ordinateur, ceci a besoin ainsi de logiciels capables d'interpréter l'information reçue et la transformer en musique. L'étude de cas de cet article se concentre sur les logiciels de Contrôle procédural [12], plus spécifiquement sur *Max/MSP*. Dans ce type de programme, l'utilisateur définit les procédures pour effectuer la synthèse ou la manipulation du son.

### 2.3. Le mapping

Une définition générale de ce constituant de l'instrument est présentée de la manière suivante : « *The term mapping is used widely to indicate the mathematical process of relating the elements of one data set onto another.* » [4] Dans les instruments numériques, le moteur sonore est découplé de l'interface qui génère les données pour la production sonore. La mise en place d'un système de communication et traduction de ces données est donc nécessaire pour que les deux éléments puissent comprendre les messages échangés entre eux. Le *mapping* est responsable de la connexion entre les données résultant de la manipulation des interfaces et les moteurs sonores. Les valeurs possibles captées de l'interface sont adaptées aux paramètres du son (ou autre selon le type de geste) dans le moteur sonore et de la même manière dans le sens inverse. Trois types de *mapping* sont souvent mentionnés : direct, divergent et convergent. [11]

## 3. LORENZO BIANCHI HOESCH

Lorenzo Bianchi Hoesch est un artiste multidisciplinaire contemporain, travaillant principalement dans le domaine de la musique électroacoustique. Il a travaillé en tant que réalisateur en informatique musicale à l'IRCAM et ses créations ont reçu différents prix de composition. Depuis 2010, j'ai suivi son parcours à travers ses différentes productions artistiques et son métier d'enseignant (composition électroacoustique au Conservatoire de Musique de Montbéliard).

L'analyse de l'instrument de Bianchi Hoesch a débuté après une rencontre avec l'artiste à l'IRCAM en 2016, lorsque j'étais intéressé au parcours d'un musicien im-

provisateur, avec lequel j'avais en commun le développement d'un instrument de musique numérique en *Max/MSP*.

J'ai vu l'instrument numérique *Elle* en performance à plusieurs reprises, notamment dans le projet *Luminescence* en 2019 à Marseille, le projet *99* à Royaumont en 2016, le projet solo *Novosibirsk* au GRM à Paris en 2014 ou le trio avec François Couturier et Jean-Marc Larché en 2010<sup>2</sup>. Après m'avoir fourni son instrument, Bianchi Hoesch m'en a expliqué le fonctionnement au cours de plusieurs entretiens. Dans les prochains paragraphes, j'en propose donc une analyse.

#### 4. L'INSTRUMENT DE MUSIQUE *ELLE*

L'instrument numérique créé par Bianchi Hoesch est nommé *Elle*, d'après – comme il m'explique – la première lettre de son prénom « L ». Le moteur sonore est développé en *Max/MSP* : ses premières versions ont été construites dans la version 4 du logiciel et l'évolution de ce moteur sonore accompagne l'évolution du logiciel. Bianchi adapte le moteur à chaque nouvelle version de *Max/MSP* pour que tous les modules restent fonctionnels et parfois il intègre les innovations du logiciel à son moteur. Bianchi Hoesch utilise ce système pour de multiples situations artistiques où, selon ses besoins, il change et adapte l'outil au contexte musical. Le système complet intègre des modules qui peuvent être utiles pour la composition en studio et des modules réalisés pour être joués dans les performances live. Si besoin, des extensions peuvent être adjointes à son instrument pour augmenter les possibilités sonores. Bianchi Hoesch utilise *Elle* pour ses performances en solo, mais aussi lors de son travail avec d'autres instrumentistes, où il a alors la possibilité de combiner le traitement en temps réel du son capté et produit par les autres musiciens avec le son produit par son propre instrument.

J'ai eu la possibilité d'analyser quarante-trois versions du système *Elle* (de 2009 à 2017) et ainsi de noter les différences entre chacune des utilisations du système. Bianchi Hoesch sauvegarde une nouvelle version pour chaque performance avec la configuration qu'il a utilisée à cette occasion. Des nouvelles versions sont créées aussi pour chaque nouveau changement dans la structure, ou chaque ajout des modules. Ceci offre au chercheur une archive riche pour documenter l'évolution d'une pensée instrumentale.

Les performances de Bianchi étant toujours différentes, il peut s'avérer qu'une configuration spécifique doit être mise en place. Par exemple, pour le traitement en temps-réel des instruments (ex. le projet *99*), le compositeur ajoute des filtres et des potentiomètres de *gain* pour contrôler les entrées sonores dans *Elle*. Pour la performance solo (ex. le concert au GRM), il privilégie l'utilisation des modules d'échantillonnage et de synthèse sonore. Dans la comparaison entre les diffé-

rentes versions analysées d'*Elle*, j'ai vérifié que les modules internes ne sont pas toujours les mêmes, mais ils sont choisis au fur et à mesure des besoins du compositeur.

En effet, lors des répétitions, Bianchi Hoesch fige les modules qui seront utilisés pour un spectacle donné et enlève les modules qu'il n'utilisera pas. Cela permet de gagner de l'espace de computation et de simplifier l'utilisation des interfaces gestuelles. En observant de près l'évolution de son dispositif, j'ai pu voir comment Bianchi Hoesch a intégré les modules de traitement sonore et les interfaces gestuelles pour la manipulation du son.

Dans cet article je vais me concentrer sur la description des composants principaux qui, à mon avis, montrent plus clairement la pensée créative et la qualité sonore cherchée par le musicien. Je renvoie le lecteur à ma thèse [3] pour une analyse plus détaillée de l'instrument numérique de Bianchi Hoesch.

##### 4.1. Le système de routage du son

Une des caractéristiques les plus intéressantes de cet instrument électroacoustique est le système de routage du son entre les différents modules de génération et de traitement du son qui composent *Elle*. C'est-à-dire, la façon dont le son est reçu et renvoyé entre les différents modules. Le système est construit de façon à ce qu'à l'intérieur de chaque module, le musicien puisse régler la quantité de son provenant de chacun des autres modules, ainsi que le son d'entrée externe pour son traitement. Cela permet une infinité d'enchaînements de transformations sonores à partir d'un seul signal audio. Dans la Figure 2, on peut voir le tableau d'entrées de routage réglable qui permet d'adapter la quantité de son de chaque module. Ce tableau est présent dans tous les modules du système avec la même configuration d'ordre et de couleurs. Cela permet l'apprentissage et le repérage de l'état des entrées audio dans le module.



**Figure 2.** Module pour la définition de la quantité de son des entrées dans un module.

<sup>2</sup> Plus d'informations sur les projets de Lorenzo Bianchi Hoesch dans son site web personnel : voir [www.lorbi.info/wp/?page\\_id=715](http://www.lorbi.info/wp/?page_id=715).

## 4.2. Les modules audio

### 4.2.1. Modules de synthèse sonore

Bianchi Hoesch conçoit deux modules audio similaires pour la génération sonore. Nommés *LDist-follow* et *LDistC*, ces modules prennent une banque d'oscillateurs interpolés (*interpolated oscillator bank*, objet *ioscbank~*<sup>3</sup>) pour créer un nuage de fréquences autour d'une fréquence centrale. Dans le cas du module *LDist-follow*, celle-ci est acquise à partir d'un signal d'entrée avec l'objet *sigmund~*. Le nuage suit cette fréquence. À partir du module transversal d'entrée, le musicien choisit la source pour l'obtention de la fréquence centrale. Pour créer cette sensation de nuage, le module permet l'interpolation entre les valeurs détectées et ajoute une variation à partir de valeurs aléatoires autour de la valeur de la fréquence détectée. Si la fréquence détectée ne change pas, le système donne à ce nuage la « liberté » de voyager progressivement autour d'autres fréquences (objet *drunk*). L'effet de nuage est obtenu également à travers la réverbération des sons des oscillateurs. L'utilisateur peut choisir le nombre d'oscillateurs, le nombre de *samples* pour lesquels les *smoothing* de fréquence et d'amplitude sont effectués et la vitesse d'interpolation (ms). Il est aussi possible de transposer l'onde de texture dense résultant de cette banque d'oscillateurs.

Bianchi Hoesch crée un deuxième système similaire à l'intérieur pour créer cette sorte de nuage autour d'une fréquence. Dans ce cas du module *LDistC*, la fréquence centrale n'est pas détectée par le module, mais c'est l'utilisateur qui l'introduit. Ceci est réalisé grâce au contrôleur *Pen Wacom*. Tout le reste du système reste inchangé. La position sur l'axe Y de la *Pen Wacom* contrôle la fréquence centrale des oscillateurs, et la position sur l'axe X contrôle le pourcentage de son qui sort avec (ou non) la réverbération (*wet* et *dry*).

Un autre module pour la synthèse du son très utilisé par Bianchi Hoesch est le *LGenSynth*. L'interface minimale de ce module cache la complexité de la programmation qu'il contient. Le module *gen~* est utilisé pour créer un patch de plus bas niveau que dans la programmation habituelle de *Max/MSP*. L'interface de ce module présente simplement le module de spatialisation, un champ pour choisir un *preset* et deux potentiomètres pour définir l'amplitude sonore de sortie et la quantité de réverbération qui sera appliquée au son. Pourtant, le module est très efficace dans la production des sons bruyants. Le choix du *preset* démarre le système qui prend 10 ms de bruit (avec l'objet *noise~*) et réalise des opérations de délai, de rétroalimentation et de filtrage du son pour obtenir le son résultant. Deux sous-modules *gen~* identiques forment ce synthétiseur où le son résultant de chacun devient le son d'entrée pour l'autre. La gestion des valeurs pour ces opérations est réalisée à partir de l'objet *nodes* et contrôlée par l'interface gestuelle. De la manipulation de cet objet composé par

quatre objets *nodes*, un vecteur est créé avec les distances du curseur aux *nodes*. Ces valeurs vont servir à transformer les paramètres du délai et du filtrage pour avoir une diversité dans le son produit. Le module *gen~* est composé de différents délais enchainés entre eux, rendant le son avec la rétroalimentation et le filtrage toujours différent et imprévisible. C'est donc le mouvement du curseur dans l'objet *nodes* qui provoque la transformation sonore. Cet objet, à deux dimensions, est manipulé par le contrôleur *Wacom* qui est parfaitement adapté à cet objet.

### 4.2.2. Scrub : Lscrub

Venant de l'anglais, *scrub* signifie froter, balayer avec une brosse. Dans ce module, il s'agit de « froter » à travers la longueur d'un échantillon sonore, tel un disque parcouru par l'aiguille d'une platine. Ce module est l'un des plus importants dans le dispositif de Bianchi Hoesch, car il est très souvent associé aux interfaces gestuelles les plus utilisées par le musicien. À travers l'interface gestuelle ou même en survolant avec une souris sur l'objet *waveform~*, le musicien peut parcourir l'échantillon sonore qui est joué dans le point de contact de l'objet. Ce module utilise les objets *superVP* développés par l'IRCAM. Plus précisément, il utilise l'objet *SuperVP.scrub~* qui ré-synthétise le son de la mémoire tampon. L'utilisateur a la possibilité de choisir entre quatre échantillons sonores qui sont tirés d'un dossier en temps réel en utilisant les touches u, i, o et p. On peut voir dans la Figure 3 que l'utilisateur peut contrôler la transposition de l'échantillon, la transformation de l'enveloppe spectrale, la quantité de partiels harmoniques (sinusoïdes) et les partiels inharmoniques (noise).



Figure 3. Module Lscrub pour jouer avec un échantillon sonore

## 4.3. Analyse générale des modules du moteur sonore

L'une des caractéristiques flagrantes de l'instrument *Elle* réside dans le fait qu'il mélange à la fois des modules dont Bianchi Hoesch est l'auteur ainsi que des adaptations de modules créés par d'autres artistes ou luthiers numériques. Lors de la construction d'un module, Bianchi Hoesch effectue une recherche sur les options disponibles et réalise des tests de façon intensive pour trouver la sonorité qu'il cherche. L'importance est dans la qualité sonore atteinte, et il peut faire évoluer les patches d'autres luthiers et les adapter à la sonorité qu'il souhaite construire. Par exemple, pour les trajectoires sonores, Bianchi Hoesch a commencé par construire lui-même un module de spatialisation. Il a ensuite adapté le *spat*, un module de spatialisation beaucoup plus puissant

<sup>3</sup> Tous les objets mentionnés proviennent du logiciel Max/MSP.

conçu par l'IRCAM. Il a ainsi pu intégrer des possibilités techniques que son module n'avait pas.

En observant la construction des modules, on constate que celle-ci n'est pas uniforme, même si des sous-modules sont transversaux à tous les modules. L'accès aux paramètres, par exemple, est très différent d'un module à l'autre. Bianchi conçoit des modules où tous les paramètres sont accessibles et peuvent être modifiés ; dans d'autres cas, comme celui du *LGenSynth* analysé avant, le choix du *preset* sert à démarrer le processus et les paramètres disponibles à la manipulation sont simplement le contrôle du volume de sortie et l'ajout de la réverbération. Aucun autre paramètre ne peut être manipulé ou n'est accessible via le module. L'utilisation des interfaces gestuelles qui gèrent les paramètres explique cette option dans la construction des modules. Certains modules d'effets n'ont pas d'interface gestuelle associée et l'accès aux paramètres est réalisé directement dans le module, ce qui oblige à rendre cet accès plus explicite. Ces différences dans la conception des modules sont dues à un objectif différent, selon qu'il s'agisse du travail de composition ou d'une performance improvisée. Ainsi, pour la composition, le détail devient plus important et les paramètres doivent être facilement accessibles. Dans l'improvisation, il s'agit de privilégier le geste instrumental et le son résultant.

*Elle* évolue avec l'introduction de nouveaux modules et l'abandon de certains, ainsi qu'avec l'utilisation de nouvelles interfaces gestuelles qui donnent d'autres possibilités.

On peut remarquer l'intérêt particulier de Bianchi Hoesch pour l'utilisation des effets qui déconstruisent le son. Un exemple évident est l'effet de délai avec la ligne en retard dans différents modules, qu'il utilise pour itérer ou décomposer un son d'entrée. En tant que générateur de son, mais toujours avec la ligne en retard comme base de manipulation sonore, on trouve aussi le *LGenSynth*. De l'analyse des trois modules, on voit que le son est en transformation constante. Le compositeur introduit dans ces modules des fonctionnalités qu'il ne peut (ou ne veut) pas contrôler, tels que les itérations variables dans les délais. Bianchi souhaite créer des sonorités toujours différentes, mais avec les mêmes caractéristiques, car le son d'entrée reste le même. Une manière d'aboutir à cela est l'utilisation des fonctions qui génèrent des valeurs aléatoires qu'on trouve dans plusieurs modules. Dans le cas de *LGenSynth*, il n'y a pas vraiment de fonctions qui créent de l'aléatoire, mais l'enchaînement des délais et des filtres rendent presque impossible le fait de savoir à l'avance à quoi ressemblera le son résultant.

On note également une évolution dans la manière de produire les sons. Dans les premières versions de son instrument, Bianchi utilisait un seul échantillonneur pour le déclenchement des échantillons sonores, puis il en a progressivement ajouté d'autres. Dans la dernière version à laquelle j'ai eu accès, il utilise vingt-quatre échantillonneurs destinés à être déclenchés par ses nouveaux contrôleurs. D'autre part, il ajoute des modules de synthèse sonore pour produire le son (*LGenSynth*), ce

qui nous amène à conclure qu'il cherche progressivement à trouver une indépendance plus grande vis-à-vis des sons d'entrée (du traitement sonore). Cela lui permet d'être plus facilement indépendant des sources sonores externes et de faire, par exemple, des performances en solo. Dans le cas de la musique mixte, cela lui donne la possibilité de créer un dialogue entre les sons qu'il produit et le traitement appliqué aux instruments.

#### 4.4. Les interfaces gestuelles

Bianchi Hoesch a questionné la performance avec les instruments numériques au cours de sa carrière. Il a longuement testé une multitude de contrôleurs disponibles sur le marché (et dans le *DIY* [5]), lors de performances et en studio, pour trouver la configuration où le rapport entre le geste et le son lui convenant le mieux. J'ai parcouru les quarante-trois versions d'*Elle* et il est important de mettre en évidence la quantité de contrôleurs que Bianchi Hoesch a utilisé au cours des années : j'ai remarqué l'utilisation de treize contrôleurs qui ont été adaptés à *Elle*. Par exemple, si dans le trio avec François Couturier et Jean-Marc Larché mentionné auparavant il utilisait des contrôleurs *Wiimote*, dans son concert au GRM (Paris, 2014), il est passé à la *Wacom* et la *BCF2000* ou l'*AkaiMidiMix*. Cet artiste cherche donc dans les outils numériques une tension instrumentale équivalente à celle qu'on peut trouver avec les instruments acoustiques. Le choix des interfaces est réalisé par la relation entre la tension (ou expressivité) instrumentale que le contrôleur permet et la faisabilité des gestes qui conviennent à l'artiste. Bianchi Hoesch préfère des interfaces où il y a un contact physique avec cet objet.

Par l'analyse des dernières versions d'*Elle*, on observe que le jeu instrumental de Bianchi Hoesch est effectué par l'utilisation simultanée de plusieurs interfaces gestuelles lors de ses performances. Il décide d'en utiliser certaines plutôt que d'autres selon les situations : il peut s'agir d'une nécessité d'expressivité, de confort et de confiance dans une interface déterminée, mais aussi de leur disponibilité. Une question pertinente à ce point serait de savoir comment adapter à chaque fois le système à chaque interface. Autrement dit : comment créer l'unité de *mapping* pour chaque interface ? Est-ce que les modules concernés évoluent avec les possibilités du contrôleur où est-ce le lien instrumental qui s'adapte à la configuration du module ? Avec *Elle*, dans la majorité des nouvelles situations, le contrôleur est dédié à un module spécifique, sans que des changements y soient effectués. Chaque contrôleur a ses caractéristiques spécifiques et un changement dans un module implique la redéfinition et le réapprentissage des modes de jeu. Pensons au changement de l'interaction pour le module *LScrub* de l'interface *Wacom* à la *LeapMotion* : dans le premier cas le « balayage » est effectué avec le stylo, dans le deuxième cas il est réalisé avec la position de la paume de la main et les paramètres avec les doigts. Il s'agit de deux typologies différentes de contrôleurs al-



ternatifs [9] : respectivement un contrôleur tactile et un contrôleur immersif.

Il est difficile d'effectuer des gestes équivalents sur deux interfaces où les données sont acquises de manière si différente. Il faudrait beaucoup d'entraînement pour y parvenir. Néanmoins, on peut profiter de cette contingence. Avec le même module, on pourra obtenir une nouvelle sonorité, car un geste différent rendra le son différent. Bianchi Hoesch, effectivement, cherche ces particularités dans les différents contrôleurs qu'il expérimente. Ensuite, il travaille pour obtenir le geste expert avec les différentes interfaces. C'est le résultat sonore et le rapport entre sa personnalité et le geste instrumental qui vont définir si le changement d'interface se révèle intéressant.

Il est possible d'utiliser plusieurs interfaces en même temps avec *Max/MSP*. Pour cela, il suffit de bien choisir l'interface pour établir la relation entre le *mapping* et le module correspondant. De même, il est possible d'utiliser plusieurs modules audio avec la même interface gestuelle en faisant un *mapping* divergent de la façon qui s'ajuste le mieux au module. On remarque cette utilisation par Bianchi Hoesch, par exemple avec la *Wacom* où le même geste manipule le module *LDistC* et le *LGenSynth*.

Je vais maintenant analyser la façon d'utiliser les interfaces gestuelles dans le système *Elle*. On se concentrera sur les interfaces les plus utilisées qui permettent des modes d'expressivité plus amples que de simples états on/off ou des potentiomètres qui varient entre 0-1. On analysera donc les contrôleurs alternatifs tactiles : la *Wacom*, et le *QuNeo*. Pour une description des contrôleurs alternatifs immersifs ou mixtes utilisés par Bianchi Hoesch, *Nintendo Wii* et *LeapMotion*, je renvoie à ma thèse [3].

#### 4.4.1. Pen Wacom

Ce contrôleur est composé d'un stylo et d'un pad tactile. L'utilisateur fait un balayage avec le stylo sur le pad pour obtenir des valeurs pour les dimensions X et Y et utilise la pression du stylo sur le pad tactile pour acquérir les données de la dimension Z. L'interface *Pen Wacom* a été conçue pour les professionnels qui ont besoin de travailler avec l'écriture ou le dessin directement sur un ordinateur. Il a été adapté pour être utilisé en tant que contrôleur pour *Max/MSP* [2] dans le but d'explorer de nouvelles interfaces pour la création musicale. Il s'agit de profiter d'un « geste expert » dont parle Risset [10] dans la maîtrise d'un instrument de musique. En effet, la manipulation des crayons et puis des stylos est réalisée depuis l'âge de 3 ou 4 ans, quand l'enfant commence à dessiner et puis à écrire avec ce type d'outil. L'adaptation du geste déjà expert, de l'écriture au stylo à la musique a été profitée par les luthiers numériques. La *Pen Wacom* est donc un contrôleur dont le luthier numérique peut attendre une grande précision d'utilisation.

En analysant les différentes versions d'*Elle*, on s'aperçoit que ce contrôleur a été expérimenté avec plu-

sieurs modules. On a mentionné auparavant le *LScrub*, mais d'autres sont utilisés tels que le *LDistC* ou le *LGenSynth*. Le contrôleur peut être bien sûr utilisé simultanément dans plusieurs modules. Le *mapping* divergent peut être fait de manière à ce que le geste manipule différents paramètres de différents modules. Dans une dernière version à laquelle j'ai eu accès, la *Pen Wacom* avait la possibilité de contrôler les deux modules que je viens de citer. Le même geste dans la surface du contrôleur *Wacom* déclenche les sons de ces modules. Le *mapping* vers les deux modules est appliqué différemment. En regardant ce lien instrumental, dans le *LDistC*, le geste est utilisé pour manipuler la quantité du son direct, le son réverbéré et la fréquence centrale. La pression du stylo est « mappée » directement à l'amplitude du son : avant l'appui avec le stylo dans la surface la pression est nulle donc il n'y a pas de son ; il y a une relation directe entre la fréquence centrale et l'axe Y du contrôleur allant de 9 Hz à 12 544 Hz ; finalement, pour la réverbération Bianchi crée un niveau intermédiaire de calcul pour définir la quantité de son réverbéré. Ce niveau intermédiaire permet d'avoir une courbe logarithmique au lieu d'une fonction directe. Le contrôleur dans le cas de *LGenSynth* est utilisé pour manipuler une fonction de probabilité réalisée avec l'objet *nodes* de *Max/MSP*. Encore une fois, le *mapping* est divisé en deux niveaux. Les coordonnées XY de la *Wacom* sont transformés avec cette fonction en quatre valeurs qui deviendront les paramètres de la synthèse.

Dans le deuxième cas d'utilisation de la *Wacom*, le geste change directement les paramètres des deux synthétiseurs. Bianchi Hoesch donne la possibilité de choisir les modules qui seront contrôlés par la *Wacom* en associant des boutons du contrôleur pour associer ou dissocier les modules à son geste. L'utilisation de la *Wacom* est présentée dans une performance à la Gaité Lyrique en 2012<sup>4</sup>. Le compositeur parle de sa performance sur le site web des projets des utilisateurs de *Max/MSP* : « *All the electronic music is composed and performed with Max/MSP. In this particular case the patch that I'm controlling with the Wacom interface is a Gen~ patch that uses several delay lines of very small amounts of samples. there's also a little moment where, with the same Wacom, I'm using a superVP.ring patch. Enjoy* »<sup>5</sup>. On n'a pas relevé l'utilisation de l'objet *superVP.ring* dans l'instrument de Bianchi Hoesch, ce qui veut dire qu'il ajoute des objets à certaines occasions pour créer ponctuellement des effets.

#### 4.4.2. QuNeo

Il existe plusieurs contrôleurs MIDI avec des pads qui fonctionnent comme des boutons variant entre 0 et 1. Le *QuNeo* a ajouté deux fonctionnalités intéressantes à cette simplicité des pads. Il permet de détecter la pression du doigt sur le pad et de relever les coordonnées

<sup>4</sup> Voir [www.youtube.com/watch?v=0QqQc-KW\\_gk](http://www.youtube.com/watch?v=0QqQc-KW_gk).

<sup>5</sup> « Project: Live at Gaité Lyrique », Cycling'74. [Voir [cycling74.com/projects/live-at-gaite-lyrique/](http://cycling74.com/projects/live-at-gaite-lyrique/), consulté le 22 juillet 2018.]

XY du point pressé sur le pad. Ainsi à partir du pad présentant un seul paramètre avec simplement deux valeurs, *QuNeo* a ajouté les dimensions habituelles du MIDI présentées dans d'autres objets tels que le potentiomètre (les valeurs de 0-127).

Le *QuNeo* est utilisé dans *Elle* surtout comme interface gestuelle pour la manipulation des échantillons. Il est utilisé aussi pour le module *Lstretcher*, pour la réverbération et le reverse. Pour les échantillons, chaque pad du *QuNeo* a son interface de *mapping* pour l'envoi des coordonnées XY, de la pression du doigt et du *slideout*. Un *mapping* direct est utilisé avec les variables des modules. Bianchi Hoesch assigne les coordonnées des axes aux paramètres de position du son dans la spatialisation du système (actuellement dans le *spat* de l'IRCAM) ; la pression du doigt contrôle l'amplitude de l'échantillon et la vitesse de lecture de l'échantillon.

#### 4.5. Les conséquences sonores des choix de lutherie

L'analyse de l'instrument de Bianchi Hoesch peut nous fournir beaucoup d'informations sur le rendu sonore de l'instrument et sur la manière dont le musicien pense le son et la performance. L'étude des modules de son moteur sonore montre qu'ils sont fabriqués de manière à ce que le son puisse avoir une évolution continue. Effectivement, la répétition est constamment évitée et le musicien cherche des relations non-linéaires dans le son à différents moments. L'imbrication sonore rendue possible par son système de routage du son confirme l'observation selon laquelle chaque module peut devenir la source sonore d'un autre module. La complexité de sa musique montre qu'il explore cette caractéristique de l'instrument lui permettant de composer le son en le stratifiant. Par exemple, l'utilisation de deux modules en même temps avec la même interface gestuelle provoque cette superposition de deux strates avec le même geste. À cela on ajoute la morphologie interne du son qui dans le cas de Bianchi Hoesch a souvent un spectre bien large. Ce sont des caractéristiques qu'il cherche pour obtenir une sorte de polyrythmie entre les différentes strates sonores qui sont diffusées en même temps. Cette envie d'une évolution continue est bien visible dans la manière dont il a construit les modules. Par exemple, dans le *LDelay* la rétroalimentation des itérations avec la lecture continue provoque une transformation du son en continu.

Une autre caractéristique qu'on peut retirer de l'analyse de cet instrument est le travail sur le matériau sonore. Les modules utilisés pour la performance ne sont pas pensés pour que les paramètres soient manipulés dans le détail. Par exemple, les valeurs de la rétroalimentation ou des filtres dans *LGenSynth* ne sont pas accessibles et configurables. Le *mapping* divergent utilisé encapsule l'accès à ces données et tous les calculs réalisés. L'utilisateur est donc censé découvrir par le geste les possibilités instrumentales du module. Toutefois, la transformation continue du son rend cette tâche plus difficile. On conclut donc que cela n'est pas l'intention de Bianchi Hoesch, qui recherche une sonori-

té spécifique dont le geste favorise davantage la transformation récurrente du son.

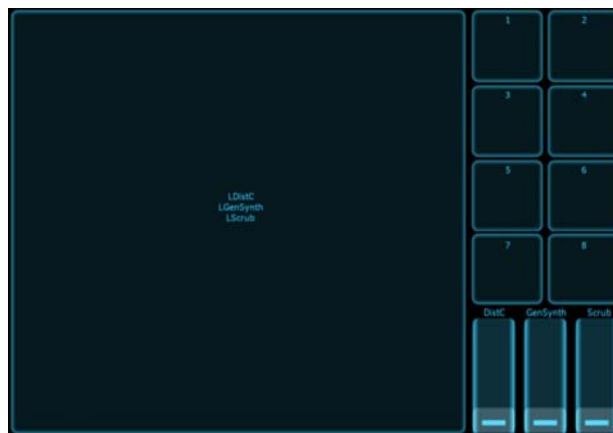


Figure 4. Simulation des interfaces gestuelles de Bianchi Hoesch (Wacom et QuNeo)

Le son n'est pas élaboré dans le détail de la progression motivique, comme on peut l'observer dans les musiques de recherche contemporaines, mais il est réalisé par des blocs où la densité sonore est importante. Le nombre d'oscillateurs (90) dans les modules *LDistC* atteste la densité souhaitée par Bianchi Hoesch. Cette densité sonore est aussi remarquée par le type d'échantillons sonores utilisés. Dans *LGenSynth*, la synthèse sonore est réalisée à partir d'un petit échantillon de bruit blanc. Le résultat du traitement repose sur cette densité sonore de base.

#### 4.6. Simulation des interfaces gestuelles d'Elle

Pour vérifier ces aspects qu'on vient d'expliquer, j'ai construit une interface gestuelle pour explorer le moteur sonore d'*Elle*<sup>6</sup>. J'ai utilisé l'*iPad* avec l'application *Liine Lemur* comme interface gestuelle. Dans la mise en place de cette simulation, j'ai été confronté à une différence entre l'ensemble des données envoyées par les interfaces de Bianchi Hoesch et celles envoyées par la mienne. Cet exemple montre ainsi la problématique de la création du lien instrumental (*mapping*) et de l'adaptation (ou évolution) vers une nouvelle interface

J'espère montrer comment les gestes utilisés par Bianchi Hoesch, avec ses interfaces gestuelles, construisent des sonorités très complexes et non linéaires. La Figure 4 représente l'interface de jeu que j'ai construit pour reproduire deux interfaces gestuelles : la *Wacom* et le *QuNeo*. J'ai projeté le pad tactile de la *Wacom* et huit (au lieu des seize) des petits pads du *QuNeo*. J'ai ajouté trois potentiomètres verticaux dédiés à l'amplitude des modules que Bianchi Hoesch utilise avec l'*AkaiMidiMix*. Dans cet essai on perd une dimension expressive et importante présente dans ces deux contrôleurs : la pression exercée sur le pad tactile, inexistant

<sup>6</sup> Nous renvoyons le lecteur à la vidéo sur la plateforme *YouTube* pour montrer le résultat sonore de cette simulation : voir [youtu.be/iIGc221EiWo](https://youtu.be/iIGc221EiWo).

dans l'*iPad*. Pour compenser cette dimension perdue, j'ai construit une pression artificielle. La pression est donc obtenue par le nombre de doigts utilisés dans l'espace de jeu. J'ai profité du fait que dans la *Wacom* et le *QuNeo*, les pads sont utilisés avec un seul stylo ou un seul doigt. Ainsi, si la pression correspondait à un intervalle entre 0 et 1, sans doigts sur la surface correspondra à la pression 0 et avec cinq doigts correspondra à la pression 1. Cependant, on ne peut pas obtenir toutes les valeurs intermédiaires directement. Pour que la pression ne reste pas figée dans une des cinq positions possibles, j'ai ajouté une petite valeur aléatoire (entre 0 et 1). Le *mapping* original des axes XY reste inchangé. Les modules que j'ai utilisés pour cette simulation sont le *LGenSynth*, le *LDistC* et le *LScrub* pour le grand espace de jeu. Pour les pads, j'ai utilisé les échantillonneurs du panneau principal. À noter que la taille des pads ne permet pas d'insérer plus d'un doigt dans l'espace de jeu. Nous avons donc rejeté la pression dans les échantillonneurs. Ainsi c'est l'axe Y qui permet la manipulation de l'amplitude de la reproduction des échantillons.

La création et expérimentation avec cette interface n'ont pas pour objectif d'obtenir un geste « expert » dans l'utilisation d'*Elle*, mais d'illustrer musicalement la complexité spectrale et la non-linéarité que ce système engendre. D'ailleurs, par une analyse fine de la relation entre le geste et le résultat sonore, il est clair que je n'ai pas la virtuosité dans l'utilisation de chaque module. En effet, la non-linéarité des modules a pour conséquence qu'on n'a pas la certitude du son que résultera du geste. La superposition des modules avec le même geste rend cette incertitude exponentielle. Ce sentiment est évident lorsqu'on baisse l'amplitude d'un module et le son résultant des autres modules est différent de ce qu'on attend.

D'une part cette caractéristique rend difficile la conception d'une macroforme pour certains paramètres, car le résultat à chaque moment est imprévisible. Par exemple, le cheminement des structures dans le détail de la composition du spectre s'avère compliqué à achever. D'une autre part, cette imprédictibilité nous oblige à retourner aux origines de l'improvisation, à réagir à l'imprévu, l'instant même du son et suivre chaque moment de la performance. Dans cette expérience, j'ai réussi à trouver une forme à partir du choix des modules à utiliser à chaque moment (à travers l'utilisation des potentiomètres d'amplitude). Ainsi, au lieu de chercher une évolution contrôlée à l'intérieur de chaque module (ce qui s'est révélé impossible), j'ai développé l'improvisation par des blocs en faisant apparaître ou disparaître la sonorité de chaque module ou des échantillons.

L'utilisation des échantillons change cette imprévisibilité, car le son stocké dans la mémoire tampon est figé. Dans le cas des échantillons de courte durée (tel que nous avons utilisé), le résultat devient rapidement très répétitif, ce qui contraste avec les autres modules. Toutefois, *Elle* a des moyens de faire de sorte à ce que le son soit toujours différent comme on a remarqué auparavant. Ainsi, pour être cohérent avec l'utilisation des

autres modules, j'ai ajouté à l'échantillonneur le module de délai. L'audio résultant des pads est envoyé à ce module que j'ai utilisé avec les temps de délai qui varient en continuation et la lecture à vitesse variable des deux mémoires tampon rend l'utilisation des échantillons imprévisible de la même manière que les modules *LDistC* et *LGenSynth*.

Lors de ma rencontre avec Bianchi Hoesch, celui-ci m'explique que dans son travail futur il aimerait bien pouvoir changer cette identité sonore ancrée dans l'instrument pour une autre plus contrastée, qu'il appelle une « esthétique de simplicité ». Si dans certains modules il suffit de redéfinir les valeurs des *presets* (par exemple réduire le nombre d'oscillateurs dans *LDistC*), d'autres modules (tels que le *LGenSynth*) devront être repensés.

## 5. CONCLUSIONS

Dans cet article j'ai proposé une analyse détaillée de l'instrument de musique numérique du compositeur Lorenzo Bianchi Hoesch, pour mieux comprendre la musique exécutée par ce musicien. Il s'agit d'observer les outils de création musicale en premier, dans une démarche complémentaire à l'analyse isolée du produit sonore final. Car, si on en reste à celui-ci, dans le cas de ce compositeur, il est compliqué de déduire que l'évolution d'une texture sonore est la conséquence d'un processus aléatoire. À l'inverse, en regardant du côté du code du moteur sonore nous avons immédiatement cette information.

D'un autre côté, les décisions prises dans l'implémentation des composants de l'instrument montrent comment l'instrumentiste développe son expressivité. Dans le cas de Bianchi Hoesch, on a vu qu'il utilise des interfaces gestuelles physiques, disponibles dans le marché. Ces solutions commerciales ne permettent pas d'être modifiées ; on a pu ainsi observer que ce musicien donne plus d'importance à la conception du moteur sonore et au *mapping*, pour atteindre l'expressivité souhaitée.

Avec l'analyse des modules audio de l'instrument de Bianchi Hoesch, on a pu comprendre sa façon de concevoir la musique. L'artiste recherche une évolution continue du son, qu'il réinjecte dans les modules en le transformant toujours aléatoirement. Il utilise des textures complexes avec plusieurs strates qui se modifient constamment. Tout cela est réalisé par l'emploi d'échantillons de bruit et d'un grand nombre d'oscillateurs dans la synthèse sonore. On a vu comment son système de redirection du son entre les modules permet à Bianchi Hoesch l'imbrication de plusieurs traitements simultanément. Cela permet la composition et superposition du son et de ses traitements. Par conséquent, le son qu'on entend pourrait être le résultat de transformations continues de plusieurs modules en même temps.

La façon dont les interfaces sont utilisées pour manipuler les modules est également importante pour comprendre cette complexité que le musicien cherche. Le



même geste est utilisé pour manipuler plusieurs modules audio simultanément. Un *mapping* divergent, tel que Bianchi Hoesch l'utilise, rend la relation moins évidente entre le geste et le résultat sonore.

À travers le cas de ce compositeur, il est possible de considérer que le processus créatif du musicien-luthier réside non seulement dans la façon dont il explore l'instrument en situation de performance, mais aussi dans tout le travail de lutherie numérique, réalisé au fil du temps. Car déjà dans les décisions que le luthier prend lors de la fabrication de son instrument, celui-ci opère ses choix esthétiques, et cela en relation avec les contraintes techniques de ce type d'objet.

## 6. REMERCIEMENTS

L'auteur de cet article tient à remercier Lorenzo Bianchi Hoesch pour sa disponibilité dans le partage de tout le matériel utilisé pour l'analyse de l'instrument et pour le temps accordé dans les entretiens réalisés.

## 7. RÉFÉRENCES

- [1] Bossis, B. « Écriture instrumentale, écriture de l'instrument », *Composer au XXIe siècle : pratiques, philosophies, langages et analyses*, Stévançe S. (dir.). Vrin. Paris, 2010, p. 119-135.
- [2] Couturier, J.-M. *Utilisation avancée d'interfaces graphiques dans le contrôle gestuel de processus sonores*, thèse de doctorat, Aix-Marseille 2, 2004. [[www.theses.fr/2004AIX22074](http://www.theses.fr/2004AIX22074), consulté le 16 juin 2019.]
- [3] Fernandes, J. *L'improvisation musicale électroacoustique : enjeux et problématiques du développement des technologies numériques*, thèse de doctorat, Université Paris 8, 2019. [[hal.archives-ouvertes.fr/tel-02944896](http://hal.archives-ouvertes.fr/tel-02944896).]
- [4] Hunt, A., Wanderley, M. « Mapping performer parameters to synthesis engines », *Organised Sound* 7/2 (2002), p. 97-108. [Doi: 10.1017/S1355771802002030.]
- [5] Iazzeta, F. « Performance na música experimental », *Actas do Performa' 11 : Encontros de investigação em performance*, 2011, p. 1-11.
- [6] Jordà, S. *Digital Lutherie Crafting musical computers for new musics : performance and improvisation*, thèse de doctorat, Universitat Pompeu Fabra, 2005.
- [7] Magnusson, T. *Epistemic Tools : The Phenomenology of Digital Musical Instruments*, thèse de doctorat, University of Sussex, 2009. Morreale, F., McPherson, A. « Design for Longevity: Ongoing Use of Instruments from NIME 2010-14 », *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2017, Copenhagen, Denmark, p. 192-197. [[www.nime.org/proceedings/2017/nime2017\\_paper0036.pdf](http://www.nime.org/proceedings/2017/nime2017_paper0036.pdf).]
- [8] Mulder, A. « Towards a Choice of Gestural Constraints for Instrumental Performers », *Trends in Gestural Control of Music CD-ROM*, Wanderley, M. M., BATTIER, M. (dir.). Ircam, Centre Pompidou, Paris, 2000.
- [9] Risset, J.-C. « Nouveaux gestes musicaux : quelques points de repère historiques », *Les nouveaux gestes de la musique*, Genevois, H., De Vivo, R. (dir.). Parenthèses, Marseille, 1999, p. 19-33.
- [10] Rován, J. B., Wanderley, M. M., Dubnov, S., Depalle P. « Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance », *KANSEI – The Technology of Emotion*, octobre 1997, Genes, Italy, p. 68-73. [[hal.archives-ouvertes.fr/hal-01105514](http://hal.archives-ouvertes.fr/hal-01105514), consulté le 27 juin 2019.]
- [11] Zadel, M., Scavone, G. « Laptop performance : Techniques, Tools, and a New Interface Design », *Proceedings of the International Computer Music Conference*, 2006, p. 643-648. [[hdl.handle.net/2027/spo.bbp2372.2006.132](http://hdl.handle.net/2027/spo.bbp2372.2006.132).]

Texte édité par Tom Mays